



CALORiE OPTiMiZER

Get to know more about
what we consume every day



/// iNTRoDUCTION ///

Calorie Optimizer is a practical solution for weight management. Our project employs Greedy approach to optimize daily caloric intake, prioritizing essential nutrients like protein, carbohydrates, and fiber. Tailored to individual requirements, our tool aids those on a weight loss journey by providing practical and personalized nutritional guidance.

PROBLEM STATEMENT

- The rising concern of obesity and associated health issues has prompted the need for effective weight management solutions. Many individuals face challenges in navigating the complexities of nutrition while striving for weight loss. The existing tools often lack personalization and may not consider the diverse nutritional requirements of individuals. Our project addresses this gap by presenting a Calorie Optimizer with a Greedy approach, aiming to provide a user-friendly and tailored solution for effective weight management.
- In a world grappling with obesity, our project tackles the challenge of personalized weight loss. Existing tools fall short in meeting individual nutritional needs. Enter our Calorie Optimizer: a Greedy approach to simplify weight management, ensuring users receive a tailored, practical solution for their journey to wellness



DESIGN APPROACH

The Calorie Optimizer utilizes a Greedy approach based on a modified Knapsack algorithm.

This algorithm efficiently selects food items to maximize nutrient intake while adhering to specified calorie constraints. By iteratively adding foods with the highest nutritional value relative to their calorie content, The Greedy Knapsack algorithm ensures an optimal distribution of proteins, carbohydrates, and fiber, facilitating effective weight management.

This approach strikes a balance between computational efficiency and providing users with a personalized and practical nutritional plan.

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5
6 struct FoodItem {
7     string name;
8     int calories;
9     int nutrient_value;
10 };
11
12 bool compareByNutrientValuePerCalorie(const FoodItem &a, const FoodItem &b) {
13     return (a.nutrient_value * 1.0 / a.calories) > (b.nutrient_value * 1.0 / b.calories);
14 }
15
16 vector<FoodItem> calorie_optimizer(vector<FoodItem> &food_items, int calorie_limit) {
17     sort(food_items.begin(), food_items.end(), compareByNutrientValuePerCalorie);
18
19     vector<FoodItem> selected_items;
20     int total_calories = 0;
21
22     for (const auto &item : food_items) {
23         if (total_calories + item.calories <= calorie_limit) {
24             selected_items.push_back(item);
25             total_calories += item.calories;
26         }
27     }
28
29     return selected_items;
30 }
31
32 int main() {
33     vector<FoodItem> foods = { {"Chicken Breast", 165, 25}, {"Brown Rice", 215, 5}, {"Broccoli", 55, 3} }; //Add the food items
34
35     int limit = 500; // Set your daily calorie limit
36
37     vector<FoodItem> result = calorie_optimizer(foods, limit);
38
39     cout << "Selected Food Items:" << endl;
40
41     int totalCalories = 0;
42
43     for (const auto &item : result) {
44         cout << item.name << " - Calories: " << item.calories << ", Nutrient Value: " << item.nutrient_value << endl;
45         totalCalories += item.calories;
46     }
47     cout << "Total Calories Obtained: " << totalCalories << endl;
48
49     if (totalCalories <= limit) {
50         cout << "Total calories obtained are within the daily calorie limit." << endl;
51     } else {
52         cout << "Total calories obtained exceed the daily calorie limit." << endl;
53     }
54     return 0;
55 }

```

CODE





OUTPUT



Output:

Selected Food Items:

Chicken Breast - Calories: 165, Nutrient Value: 25

Broccoli - Calories: 55, Nutrient Value: 3

Brown Rice - Calories: 215, Nutrient Value: 5

Total Calories Obtained: 435

Total calories obtained are within the daily calorie limit.

/// TIME COMPLEXITY ///

The time complexity of the provided calorie optimizer algorithm is primarily determined by the sorting operation, which is typically the dominant factor. Let (n) be the number of food items.

Time Complexity:

Sorting the food items based on nutrient value per calorie: $O(n/\log n)$.

Iterating through the sorted list to select items: $O(n)$.

So, the overall time complexity is $O(n/\log n)$.





SPACE COMPLEXITY



The space complexity is mainly influenced by the data structures used, particularly the vector storing the selected food items.

The space required for the selected_items vector - $\underline{O(n)}$ in the worst case, where n is the number of food items.

Therefore, the overall space complexity is $\rightarrow \underline{O(n)}$.

REAL LIFE USES

1. Personalized Diet Planning: The algorithm can be integrated into apps or services that offer personalized diet plans based on individual nutritional requirements, preferences, and weight management goals.
2. Nutrition Tracking Apps: Existing nutrition tracking applications can enhance their functionality by incorporating the Calorie Optimizer to suggest well-balanced meals that align with users' dietary preferences and constraints.
3. Weight Loss Programs: Weight loss programs and fitness platforms can use this tool to create tailored meal plans for users, promoting effective weight management while ensuring optimal nutrient intake.
4. Health and Wellness Services: Healthcare providers and wellness coaches can utilize the Calorie Optimizer to assist clients in achieving specific health goals, such as managing diabetes, improving heart health, or addressing nutrient deficiencies.
5. Corporate Wellness Programs: Companies aiming to promote employee well-being can integrate the Calorie Optimizer into workplace wellness initiatives, offering employees personalized nutrition guidance to support a healthier lifestyle.
6. Educational Tools: The Calorie Optimizer can serve as an educational tool in schools or nutrition workshops, helping individuals understand the importance of balanced nutrition and making informed dietary choices.